Speech Recognition of Dialectal Arabic

Rami Doas
rnd2110@columbia.edu
Columbia University

Abstract—**Automatic Speech Recognition is the process of identifying words and phrases in a spoken language and converting them into text. Recently, there has been increasing interest in speech recognition for Arabic, Dialectal Arabic in particular. This work proposes an end-to-end speech recognition system for two Arabic dialects: Egyptian and Levantine, where the system is extensible to cover other dialects upon the availability of their speech-to-text corpora. The creation of the system involves new pre-processing techniques for speech-to-text data for Dialectal Arabic, in addition to the creation of new pronunciation lexicons for Dialectal Arabic. Moreover, Since diacritics (Arabic short vowels, double-consonant markers (*shadda*) and *tanween*) are usually omitted in Arabic and always omitted in Dialectal Arabic (DA), which imposes a high degree of ambiguity, two variations of the system are investigated; where the pronunciation lexicon is either diacrizied or not. The results show that the undiacritized version of the lexicon gives a better overall WER (word error rate) of 35.3% as opposed to 39.8% in the case of a diacritized lexicon.**

I. Introduction

There has been recently enormous research on Automatic Speech Recognition (ASR), that is converting speech into text. While most of the work targets English and other common languages, research on Arabic is lacking.

Arabic is a morphologically complex language of more than 400 million native speakers, and has two main variants: Modern Standard Arabic (MSA) and Dialectal Arabic (DA). MSA is the official language and is extensively used in official media and education materials. In contrast, DA is the everyday language of Arabs and differs from MSA in its lexicon, morphology, phonology and syntax. DA has seven main dialects: Egyptian, Levantine (Lebanese, Syrian, Jordanian and Palestinian), Gulf (Saudi, Emarati, Qatari, Kuwaiti, Bahraini and Omani), Moroccan (Moroccan, Algerian, Tunisian and Libyan), Iraqi, Yemeni and Sudanese, where each dialect has several sub-dialects depending on the district. For instance, Egyptian has Cairene, Alexandrian, Upper Egyptian and Nubi as four sub-dialects. The increasing interest in the processing of DA is because it is what Arabs actually use in their communication, either in their everyday conversations or in social media.

Arabic processing is challenged by a complex affixational and inflectional system [1], where the analysis of a word requires the recognition of a large set of features such as the part-of-speech tag (e.g., noun, verb, adjective, adverb, etc.), gender, number, person and voice and clitics. This makes the word structure complex and rich. For example, the English sentence "*and they will write it for him*", could be expressed in one word in Arabic *wsyktbwhAlh* [1]

In addition to the morphological complexity, Arabic is usually written without diacritics (short vowels, double-consonant markers (*shadda*) and *tanween*), while some letters are normalized to others (e.g., Hamzated Alefs are usually written as, simply, Alef). The omission of diacritics along with letter normalization add a high degree of ambiguity as a word might have several possible analyses, which motivates context-level processing. An example of such ambiguity is the word *wjd*, which could mean several concepts depending on whether the beginning "w" is a prefix or part of the stem in addition to the actual diacritics of the stem. As a result, the word *wjd* might be interpreted as *wajad* "*he found*", *wujid* "*it was found*", *wijd* "*passion*", *wa+jid* "*and a grandfather*", *wa+jad* "*and he worked hard*". Even with the absence of affixes and the presence of diacritics, the word might still be ambiguous. For instance, the word *zahab* could mean *he went*, *gold* or the name of the Egyptian city *zahab*.

Moreover, DA does not have a standard orthography. That means the same word could be written indifferent forms depending on the perception of the writer on how to map sounds to letters. The lack of a standard introduces data sparseness and requires special processing when applying Natural Language Processing (NLP) techniques. For example, the Egyptian word that means "also" does not have an MSA cogent, and is usually written in several forms such as *brdw*, *brDw*, *brdh* and *brDh*. Another instance is the third-masculine-singular suffix, which is written as *h* (the correct MSA form) or *w*, and the third-plural suffix, which is written as *wA* (the correct MSA form) or *w*. The choice of *w* for both the third-masculine-singular and the third-plural suffixes is a common problem of ambiguity in DA.

The objective of this work is to implement an end-to-end ASR system for the most commonly used Arabic dialects: Egyptian and Levantine, where Levantine has four main sub-dialects: Lebanese, Syrian, Jordanian and Palestinian. However, one of the main aspects of the proposed system is its extensibility to include other dialects without the need

[1] Arabic transliteration in this paper is presented in the Buckwalter scheme http://www.qamus.org/transliteration.htm.

to change the core system, but only by incorporating the necessary resources such as the speech-to-text corpus and the text required to build the language model. The main contribution of this work is fourfold:

- developing an end-to-end System for DA,
- introducing techniques to tackle the limited quality of the available speech-to-text data for DA,
- creating new pronunciation lexicons for DA,
- comparing the use of a diacritized pronunciation lexicon versus an undiacritized one.

The rest of the paper is organized as follows: Section II presents the most relevant prior research on Arabic speech recognition. Section III lists the sources of the speech-to-text data and the language model data used in the proposed system. It also discusses the main drawbacks in the data that affect the quality of the speech recognition system. Section IV discusses the proposed system including the creation of the linguistic resources necessary for Arabic speech recognition. The experiments and the results of the different setups are then listed in Section V. Finally, section VI concludes and outlines possible future work.

II. Related Work

Despite the fact that Arabic is a widely spoken language, research on Arabic ASR is relatively little when compared to that of other common languages such as English and French. This is attributed to the fact that Arabic is a morphologically complex language, in addition to the limited number of Arabic resources such as corpora and language models, where the existing ones have limited quality and/or use limitations.

Most previous work on Arabic speech recognition concentrated on developing recognizers for Modern Standard Arabic (MSA), while Dialectal Arabic (DA) has received less interest, either because its lake of standards, or because of its limited resources. An early successful work on MSA was proposed in [2], where they used recurrent neural networks instead of the commonly used Hidden Markov models (HMM).

Research on MSA was then followed by a shift to DA as the importance of DA has been widely recognized especially with the increasing interest in social media. An early comprehensive study was conducted by Kirchhoff et al. [3], who investigated the recognition of DA, and studied the discrepancies between MSA and DA from the speech recognition point of view. They found that using phonetic information available in romanized as opposed to vowelless transcriptions significantly improves word error rate.

Motived by the morphological complexity of Arabic, Vergyri et al. [4] investigated the use of morphology-based language models in an ASR system for DA, which leads to a reduction in the word error rate. However, one drawback is that the full potential of the factored language models cannot be directly exploited because of the absence of decoders that support factored word representations

Another dialectal system of interest was introduced by Biadsy et al. [5]. The system does automatic identification of four Arabic dialects; Egyptian, Levantine, gulf and Iraqi, in addition to MSA. The identification of MSA gives the best results, which is attributed to the quality of the MSA resources, while the overall accuracy was 81.6% using 30s test utterances.

Other work aimed at improving the quality of Arabic speech recognition by having better modeling techniques. A notable instance is the work done by Kirchhoff et al. [6], who implemented cross-dialectal data sharing for acoustic modeling, where the use of MSA acoustic data improved the performance of Egyptian speech recognition. Another work was done by Vergyri et al. [7], who improved the acoustic modeling of DA by applying automatic diacritization techniques on the transcribed data.

The most related work was done by Ali et al. [8], where they developed a complete Kaldi[2] Receipt for Arabic speech recognition (Gale-Arabic). The best system they achieved uses DNN+MPE (Dynamic Neural Networks + Minimum Phone Error), with a word error rate (WER) of 15/8% for reports, 32.2% for conversation and 26.9% for a combination of reports and conversations, while using DNN only results in a higher WER by about 3% absolute. Although their system is developed and tested on MSA, and thus not comparable to the proposed dialectal-Arabic system in this paper, their system is easily extensible to DA upon the availability of resources.

III. Data

A. Data Sources

As stated earlier, the aim of this work is to build an ASR system for two Arabic dialects: Egyptian and Levantine, where the latter has four main sub-dialects: Lebanese, Syrian, Jordanian and Palestinian. The dialect selection is attributed to the availability of the linguistic tools and resources, speech-to-text corpora in particular.

For Egyptian, the system uses the CallHome Egyptian Arabic corpus (CallHome) (speech: [9] - text: [10]). The corpus consists of 120 transcribed telephone conversations of native speakers of Egyptian Colloquial Arabic (ECA), where all the calls have a maximum of 30 minutes, and originated in North America, and were placed to locations overseas (typically Egypt). The corpus is split into three main data sets for training, development and test containing 80, 20, and 20 conversations, respectively, for a total of 60,700 utterances. However, at the time of developing the system, only the training set was available, so 72 conversations of the training set are used for training, while the remaining eight conversations are used for testing the system.

The audio data of CallHome are 8KHz *SPH* files that contain two channels. In most of the cases, each channel corresponds to a single speaker, except in a few cases where

[2] Kaldi: http://kaldi-asr.org/

the speech of two or more speakers is recored on the same channel. However, there is no single case where the speech of two speakers intersect on the same channel.

The transcripts of CallHome are one file per conversation with time information for each utterence. However, they have a major drawback as the transcription does not use the Arabic orthography, but is phonetic-based instead. However, the corpus comes with tools that allow generating the corresponding Arabic orthography. The transcripts indicate the cases where the words are incomplete, where the audio has noise, and where the transcribers have doubts.

For levantine, the system uses Babylon Levantine Arabic corpus (Babylon) (speech and text: [11]). The corpus consists of spontaneous speech, recorded from subjects speaking in Levantine colloquial Arabic, and its transcripts. The corpus consists of 164 subjects containing a total of 75,900. For the development of the proposed system, the corpus is divided into two sets for training and testing. The training set contains 148 subject consisting of about 68,200 utterances, while the rest is used for testing the system.

The audio data of Babylon are 8KHz *WAV* files containing only one channel, where each audio file has the recording of only one speaker and one utterance.

The transcripts of Babylon have the same structure as the audio data, where there is one transcription file per audio file, and they are all in Arabic orthography with some markers, mainly to detect silence.

In addition to the speech-to-text corpora, a dialectal corpus from the Linguistic Data Consortium (LDC) of 8 million words is collected[3]. The corpus consists mainly of Egyptian text in addition to some Levantine, and is used along with the transcripts of CallHome and Babylon to generate the language model necessary for the speech recognition system.

B. Drawbacks in the Data

The currently existing speech-to-text corpora for DA, including CallHome and Babylon, involve a number of major drawbacks in the original audio and the transcripts, which requires special processing before building a speech recognition system. The drawbacks are listed below:

1. Lack of a standard Some transcripts are written in phonology-based transcripts, such as CallHome, while others use the conventional Arabic orthography, such as Babylon. Additionally, the corpora come in different structures and file formats. For example, CallHome audio files are *SPH* files of two channels, where each channel has the recording of a speech of one or more speakers, and each audio file has several utterances of a 30-minute conversation. In contrast, Babylon audio files are *WAV* files grouped into subjects, where every audio file has only one channel of one speaker, and there is a separate file for every utterance.

2. Bad Quality The audio files of both CallHome and Babylon are of a low frequency of 8 KHz. Additionally, CallHome consists of phone conversations that have noise, incomplete words and lots of cases where the transcribers have doubts regarding what is actually said. Moreover, CallHome has lots of foreign words, which confuse Arabic processing tools and language modeling.

3. Omission of Short Vowels Both CallHome (after converting to Arabic orthography) and Babylon omit the diacritics, and hence the short vowels, which affect the overall speech recognition system as the transcripts become ambiguous, where a given word might have different concepts of different pronunciations.

3. Limited Sizes So far, there is no speech-to-text corpora for DA of a large size that is adequate to build a robust speech recognition system. For instance, the total number of utterances in CallHome and Babylon is 60,700 and 75,900, respectively.

IV. Approach

A. System Structure

Figure 1 illustrates the overall process of building the speech recognition system. First, the speech-to-text corpora, CallHome and Babylon, are cleaned-up and pre-processed. The data from both resources are then merged into one resource. On the other side, the text corpus that forms the basis for the language model is cleaned-up and pre-processed. The processed transcripts and text corpus are then merged together and processed to form a language model. Additionally, a phoneme-to-grpheme lexicon is created out of the processed textual data. Next, the processed speech-to-text data runs through two modules for signal processing and feature extraction, leading to the creation of an acoustic model.

Figure 2 illustrates the testing process. As in any regular ASR system, the input speech is first captured and converted into acoustic features. These features are then processed by a decoder or a set of decoders that make use of the linguistic resources, namely; the language model, the pronunciation lexicon and the acoustic model. The models and the lexicon are exploited by the decoder or the set of decoders to estimate the probability that a given input sequence belongs to the class on which it was trained. The plausible sequences with the highest probabilities represent the final text the corresponds to the spoken input.

The next three subsections illustrate the cleanup and preprocessing steps for the training data, the creation of the linguistic resources, and building the speech recognition system.

B. Data Preparation

As mentioned in sub-section , the speech-to-text corpora lack a standard, come in bad quality and do not include short vowel information. As a consequence, it is necessary to cleanup and reprocesses the data before building the system.

[3] All of the mentioned resources are available from the Linguistic Data Consortium: www.ldc.upenn.edu

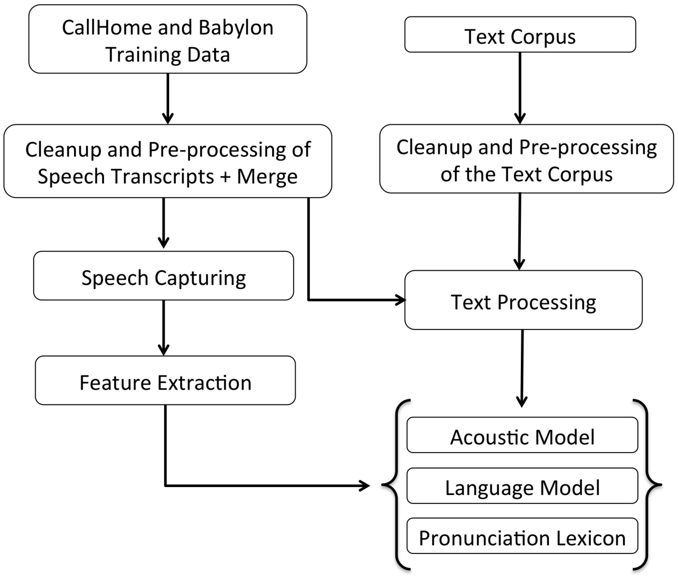

CallHome and Babylon Training Data
Text Corpus
Cleanup and Pre-processing of Speech Transcripts + Merge
Cleanup and Pre-processing of the Text Corpus
Speech Capturing
Text Processing
Feature Extraction
Acoustic Model
Language Model
Pronunciation Lexicon

Fig. 1. Building the ASR systems

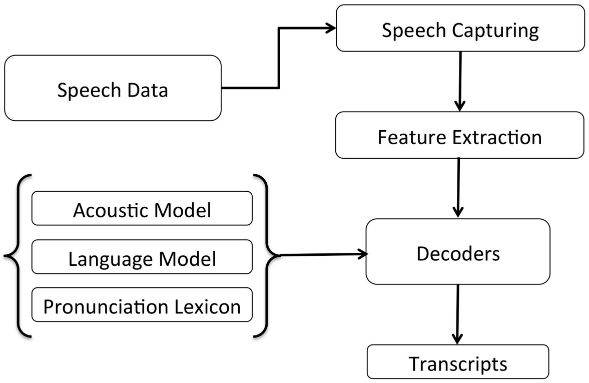


Fig. 2. Querying the ASR system

First, the CallHome audio files reconverted into the *WAV* format in order to be consistent with the Babylon ones. Next, all the utterances in CallHome that have noise, incomplete words, Non-Arabic words, or words that were transcribed with low confidence are eliminated. As a result, the number of training and test utterances decrease to 10,644 and 1,191 respectively. On the other side, the marker words, such as those indicating silence, in Babylon are eliminated, but this does not result in reducing the total number of Levantine utterances.

After cleaning up the data, all the transcripts and audio files of CallHome are cut into smaller units, where every resulting transcription file belongs to one utterance and one audio file. However, since the audio files of CallHome contain two channels, the channels are separated first before splitting further into single utterances. Although this could be managed from inside the training of the speech recognition system, it is done as part of the preprocessing phase in order to make both the Egyptian and Levantine data have the same structure as they merge together prior to building the speech recognition system.

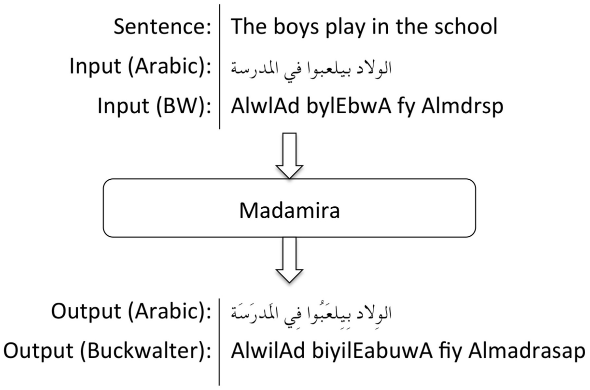
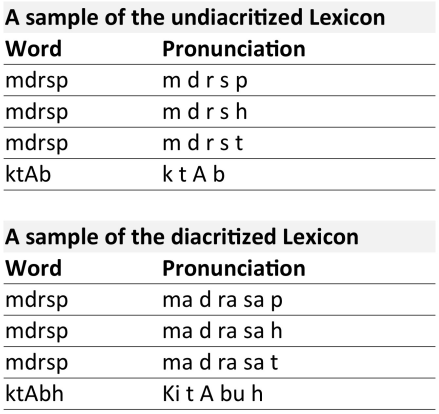

Sentence:
The boys play in the school
Input (Arabic):
الولاد بيلعبوا في المدرسة
Input (BW):
AlwlAd bylEbwA fy Almdrsp
Madamira
Output (Arabic):
الوِلاد بِيِلعَبُوا فِي المَدرَسَة
Output (Buckwalter):
AlwilAd biyilEabuwA fiy Almadrasap

Fig. 3. Madamira Example

On the other side, the textual data intended to use for the creation of the language model are cleaned up where the utterances that contain Non-Arabic words are eliminated, while the punctuation marks are removed.

Next, all the textual data (CallHome, Babylon and the textual data that form the basis for the language model) run through Madamira [12] in order to restore the diacritics (Arabic short vowels, double-consonant markers (*shadda*) and *tanween*). Madamira is a state-of-the-art system for Arabic analysis and disambiguation, where diacritization is done a part of the disambiguation process. Currently, Madamira supports MSA and Egyptian Arabic, and is built on top of Sama [13], a morphological analyzer for MSA, and Calima [14], a morphological analyzer for DA (Egyptian, so far). The analyzers produce all the possible analyses for every word in the input text, then Madamira selects the top analysis in context through feature modeling and analysis ranking. The analysis includes morphological information such as word morphs and their part-of-speech tags, lemma, gender, number, person, voice, English gloss and word diacritization. Figure 3 shows an example sentence *alwlAd bylebwA fy Almdrsp* "*The boys play in the school*", before and after adding the diacritics through Madamira. It is important to note that although Madamira is not available for Levantine, running it on Levantine text helps as Levantine has a lot in common with both MSA and Egyptian. Also, mood and case ending markers are removed whenever existing. This is because DA does not have mood and case ending, unlike MSA.



A sample of the undiacritized Lexicon

Word	Pronunciation
mdrsp	m d r s p
mdrsp	m d r s h
mdrsp	m d r s t
ktAb	k t A b

A sample of the diacritized Lexicon

Word	Pronunciation
mdrsp	ma d ra sa p
mdrsp	ma d ra sa h
mdrsp	ma d ra sa t
ktAbh	Ki t A bu h

Fig. 4. Sample entries of the pronunciation lexicons

C. *Creating the Linguistic Resources*

Upon cleaning up and preprocessing the textual data (CallHome, Babylon and the textual data that form the basis for the language model), a bi-gram language model is created based on the undiacritized text using SRILM [15]. The selection of the bi-gram model is attributed to the fact that it is significantly more efficient than a uni-gram model, while the trigram model takes more processing time and yields comparable performance.

Next, the pronunciation lexicon is created. It has all the words that appear in all the available textual data, for a total of 241,400 words. Two versions of the lexicon are developed; undiacritized and diacritized. The reason for this is to investigate the effect of diacritizing the lexicon on the overall performance of the speech recognition system. In the undiacritized version, a row contains a word and its space-separate representation. However, for the words that end with the feminine marker *p*, three rows are created. The first has the original ending, while the other two ends in *h* (which sounds similar to *p* at the end of words) and *t*. The reason for the latter is that the pronunciation of *p* changes to *t* in the case of indefiniteness. The diacritized lexicon is similar to the undiacritized one except that the pronunciation has every letter attached to its diacritic, except for the cases of long vowels and the cases where a letter is not associated with a diacritic (*sukun* cases). figure 4 lists home example entries from the undiacritized and diacritized pronunciation lexicons.

D. *Training the System*

After cleaning up and preprocessing the data and building the linguistic resources, it is time to build the speech recognition system. The system is developed using the Kaldi toolkit[4]. First, four additional files that are required for system training are created: speaker-to-gender, utterance-to-speaker, speaker-to-utterance and utterance-to-audio-file. Second, the audio data is processed, and the MFCC (Mel Frequency Cepstral Coefficients) features are generated. Next, a finite state transducer (FST) is built on top of the language model. After creating the FST, several passes run on top of each other for training, decoding and alignment. The passes follow a number of passes from the VoxForge Recipe[5], which is included in the installation of Kaldi, where they start with monophone models and end with MPE (Minimum Phone Error) ones.

In the first pass, a monophone acoustic models with delta-delta features are created and decoded, then the first set of alignments are generated based on the monophone models. The monophone models are then used to create triphone ones in a similar fashion in the second pass. This is followed by another triphone pass of a higher accuracy whereit uses delta+delta-deltas features. Next, a third triphone pass runs with LDA (Latent Dirichlet Allocation) + MLLT (Maximum Likelihood Linear Transform) features on top of the second triphone pass. Next, MMI (Maximum Mutual Information) models are trained and decoded on top of the LDA+MLLT ones. This is then repeated with a boost factor of 0.05. Finally, MPE (Minimum Phone Error) models are trained and decoded on top of the LDA+MLLT ones.

V. Experiments and Results

Next, the system is tested on the combined test sets of CallHome and Babylon, discussed in sub-section III-A, where the system goes through multiple passes as described in sub-section IV-D. The system is tested in two different setups, where the pronunciation lexicon is undiacritized in one and diacritized in the other. Table-I and table II report the word error rates (WER) after each pass when using an diacritized pronunciation lexicon and an undiacritized pronunciation lexicon, respectively.

The results show that the system that uses the undiacritized pronunciation lexicon considerably outperforms the one that uses a diacritized lexicon by an error reduction of 10.3% when using monoplane models and an error reduction of 11.3% after the sixth trephine pass that uses MPE models (Minimum Phone Error). Thus, the best system is obtained when using the undiacritized pronunciation lexicon with MPE models.

It is also noted that the WER is getting lower by performing more training and decoding passes, where the bigger jump occurs when running the first triphone pass on top of the montophone one, with error reductions of 30.1% and 31.2% in the diacritized and undiacritized setups, respectively.

Unlike what was expected, the system performs better with an undiacritized pronunciation lexicon. This can be attributed to the fact that Madamira was to able to recognize only about 74% of the words in the lexicon, and hence those words received no diacritization, which in turn results in

[4] Kaldi: http://kaldi-asr.org/

[5] VoxForge: http://www.voxforge.org/

Pass	WER %
Monophone	69.2
1st Triphone	48.4
2nd Triphone (delta-delta-deltas)	48.1
rd Triphone (LDA+MLLT)	45.4
4th Triphone (MMI)	40.3
5th Triphone (MMI with boost)	39.9
6th Triphone (MPE)	39.8

TABLE I
WER OF THE DIFFERENT DECODING STAGES WHEN USING AN DIACRITIZED PRONUNCIATION LEXICON

Pass	WER %
Monophone	62.1
1st Triphone	42.7
2nd Triphone (delta-delta-deltas)	42.3
rd Triphone (LDA+MLLT)	40.3
4th Triphone (MMI)	35.8
5th Triphone (MMI with boost)	35.4
6th Triphone (MPE)	35.3

TABLE II
WER OF THE DIFFERENT DECODING STAGES WHEN USING AN UNDIACRITIZED PRONUNCIATION LEXICON

inconsistency as the same phone might receive two different pronunciations in the lexicon (with a diacritic and without). Moreover, adding diacritics with relatively small training data might add some degree of sparseness.

Comparing to other state-of-the-art systems, the performance of the proposed system is comparable to the MSA system developed by Ali et al. [8] (the Gale-Arabic receipt), with a WER of 26.9% using DNN+MPE (Dynamic Neural Networks + Minimum Phone Error), where MSA speech recognition is a considerably easier problem to solve then for DA. Moreover, the proposed system significantly outperforms all the DA systems listed in section II which report on WER, where the WER is in excess of 40%.

VI. Conclusion and Future Work

In this paper, a system for automatic speech recognition for Dialectal Arabic is developed. The system is trained and tested on Egyptian and Levantine Arabic, but it is extensible to other dialects upon the availability of resources. Several models are tested. However, the best performance is achieved by using an undiacritized pronunciation lexicon with MPE (Minimum Phone Error) models, giving a WER of 35.3%, which outperforms the currently existing Dialectal Arabic speech recognition systems that report on WER.

It is also proved that using an undiacritized pronunciation lexicon gives better results than those of a diacritized lexicon by an error reduction of 11.3%. Additionally, efficient techniques for the cleanup and preprocessing of Dialectal Arabic text are illustrated. Also, a new pronunciation lexicon of about 241,400 words is introduced.

It is planned to develop and use bigger speech-to-text corpora, and to extend the system to work on other Arabic dialects such as Gulf and Moroccan. Also, experimenting with fully diacritized data (not only the pronunciation lexicon) is under consideration. However, this requires extensive manual work until on-the-shelf diacritization tools for the different Arabic dialects become available. Moreover, it is planned to use more advanced models such as DNN (Dynamic Neural Networks), either separate or combined with other well performing models such as MPE (Minimum Phone Error).

References

[1] N. Y. Habash, “Introduction to Arabic Natural Language Processing,” *Synthesis Lectures on Human Language Technologies*, vol. 3, no. 1, pp. 1–187, 2010.

[2] M. El Choubassi, H. El Khoury, C. J. Alagha, J. Skaf, and M. Al, “Arabic Speech Recognition using Recurrent Neural Networks,” in *IEEE Intl. Symp. Signal Processing and Information Technology ISSPIT*, 2003.

[3] K. Kirchhoff, J. Bilmes, S. Das, N. Duta, M. Egan, G. Ji, F. He, J. Henderson, D. Liu, M. Noamany *et al.*, “Novel Approaches to Arabic Speech Recognition: Report from the 2002 Johns-Hopkins Summer Workshop,” in *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP’03). 2003 IEEE International Conference on*, vol. 1. IEEE, 2003, pp. I–344.

[4] D. Vergyri, K. Kirchhoff, K. Duh, and A. Stolcke, “Morphology-Based Language Modeling for Arabic Speech Recognition,” in *INTERSPEECH*, vol. 4, 2004, pp. 2245–2248.

[5] F. Biadsy, J. Hirschberg, and N. Habash, “Spoken Arabic Dialect Identification Using Phonotactic Modeling,” in *Proceedings of the eacl 2009 workshop on computational approaches to semitic languages*. Association for Computational Linguistics, 2009, pp. 53–61.

[6] K. Kirchhoff and D. Vergyri, “Cross-Dialectal Data Sharing for Acoustic Modeling in Arabic Speech Recognition,” *Speech Communication*, vol. 46, no. 1, pp. 37–51, 2005.

[7] D. Vergyri and K. Kirchhoff, “Automatic Diacritization of Arabic for Acoustic Modeling in Speech Recognition,” in *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*, ser. Semitic ’04. Stroudsburg, PA, USA: Association for Computational Linguistics, 2004, pp. 66–73. [Online]. Available: http://dl.acm.org/citation.cfm?id=1621804.1621822

[8] “A complete kaldi recipe for building arabic speech recognition systems.”

[9] A. Canavan, G. Zipperlen, and D. Graff, “CALLHOME Egyptian Arabic Speech LDC97S45,” 1997.

[10] H. Gadalla, “CALLHOME Egyptian Arabic Transcripts LDC97T19,” 1997.

[11] BBN-Technologies, “BBN/AUB DARPA Babylon Levantine Arabic Speech and Transcripts LDC2005S08,” 2005.

[12] A. Pasha, M. Al-Badrashiny, M. T. Diab, A. El Kholy, R. Eskander, N. Habash, M. Pooleery, O. Rambow, and R. Roth, “MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic,” in *LREC*, vol. 14, 2014, pp. 1094–1101.

[13] D. Graff, M. Maamouri, B. Bouziri, S. Krouna, S. Kulick, and T. Buckwalter, “Standard Arabic Morphological Analyzer (SAMA) Version 3.1,” 2009, linguistic Data Consortium LDC2009E73.

[14] N. Habash, R. Eskander, and A. Hawwari, “A Morphological Analyzer for Egyptian Arabic,” in *Proceedings of the Twelfth Meeting of the Special Interest Group on Computational Morphology and Phonology*, Montréal, Canada, 2012, pp. 1–9.

[15] A. Stolcke, “SRILM-an extensible language modeling toolkit,” in *Interspeech*, vol. 2002, 2002, p. 2002.